## Remarks

This Application has been carefully reviewed in light of the Office Action mailed December 22, 2004. Applicants appreciate the Examiner's reconsideration of the Application. Applicants believe that all pending claims are allowable over the Examiner's rejections without amendment and respectfully submit the following remarks. Applicants respectfully request reconsideration and allowance of all pending claims.

## I. Allowable Subject Matter

Applicants note with appreciation the allowance of Claims 7-12, 23, and 25. Pursuant to M.P.E.P. § 1302.14, Applicants respectfully issue a statement commenting on the Examiner's reasons for allowance. Applicants respectfully traverse the Examiner's reasons for allowance to the extent that they are inconsistent with applicable case law, statutes, and regulations. Furthermore, Applicants do not admit to any characterization or limitation of Claims 7-12, 23, and 25 or to any characterization of a reference by the Examiner, particularly any that are inconsistent with the language of the claims considered in their entirety and including all of their constituent limitations.

## II. Applicants' Claims are Allowable over the Rejections Under 35 U.S.C. § 103

In order to establish a *prima facie* case of obviousness, three requirements must be met: (1) there must be some suggestion or motivation, either in the references themselves or in the knowledge available to one skilled in the art, to modify a reference or combine multiple references; (2) there must be a reasonable expectation of success; and (3) the prior art reference (or combination of references) must teach or suggest all of the claim limitations.

### A. Independent Claims 1, 4, and 20 are Allowable over the Proposed *Stefaniak-van Eikeren* Combination

The Examiner rejects Claims 1-2, 4-6, and 20-21 under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent 6,550,054 to Stefaniak, et al. ("*Stefaniak*") in view of U.S. Patent No. 6,618,852 to van Eikeren et al ("*van Eikeren*").

Applicants respectfully submit that the Examiner has not proven a *prima facie* case of obviousness for at least two reasons. First, assuming for the sake of argument that the

Examiner has shown the requisite teaching suggestion, or motivation in the cited references to combine or modify *Stefaniak* with *van Eikeren* in the manner the Examiner proposes, the proposed *Stefaniak-van Eikeren* combination still fails to disclose, teach, or suggest each and every element of the claimed invention. Second, Applicants respectfully submit that the Examiner has not shown the requisite teaching, suggestion, or motivation to combine or modify *Stefaniak* with *van Eikeren* in the manner the Examiner proposes.

### 1. Independent Claims 1 and 4 are Allowable

Independent Claim 1, for example, recites:

> A method for outputting data from a legacy computer system, the data output in Extensible Markup Language format, the method comprising:
>
> generating a model of the legacy computer system, the model comprising one or more incidents within one or more applications that output data;
>
> mapping the model of the legacy computer system to an Extensible Markup Language schema; and
>
> based at least on the mapping of the model of the legacy computer system to the Extensible Markup Language schema, automatically modifying the one or more applications of the legacy computer system that output data, the one or more modified applications operable to output data written using a Document Object Model from the legacy computer system in Extensible Markup Language.

At a minimum, *Stefaniak*, whether considered alone or in combination with *van Eikeren*, fails to disclose, teach, or suggest the following limitations as recited in Claim 1:

- based at least on the mapping of the model of the legacy computer system to the Extensible Markup Language schema, automatically modifying the one or more applications of the legacy computer system that output data, the one or more modified applications operable to output data written using a Document Object Model from the legacy computer system in Extensible Markup Language.

In fact, *Stefaniak* is entirely unrelated to "automatically modifying [based at least on the mapping of the model of the legacy computer system to the Extensible Markup Language schema] the one or more applications of the legacy computer system that output data, the one or more modified applications operable to output data written using a Document Object Model from the legacy computer system in Extensible Markup Language," as recited in Claim 1.

*Stefaniak* is directed to a method for representing terminal-based applications in the Unified Modeling Language. (*Stefaniak*, 1:15-17) *Stefaniak* discloses the following method: (1) transforming a terminal-based application into an application specification; (2) converting the ***application specification*** (not the legacy application) into a modeling language-based representation (e.g., UML); and (3) displaying the modeling language-based representation (of the application specification of the terminal-based application) with a graphical user interface. (*See* Abstract) Thus, *Stefaniak* merely discloses generating models of legacy applications – the system disclosed in *Stefaniak* does not modify legacy computer applications, let alone "automatically modify[] one or more applications of the legacy computer system that output data" such that the "one or more modified applications [are] operable to output data written using a Document Object Model from the legacy computer system in Extensible Markup Language," as recited in Claim 1.

First, the Examiner argues that "transforming a terminal based screen application into an application specification" discloses "based at least on the mapping of the model of the legacy computer system to the Extensible Markup Language schema, automatically modifying the one or more applications of the legacy computer system that output data," as recited in Claim 1. Applicants respectfully submit that *Stefaniak* fails to support this interpretation. *Stefaniak* discloses a system that describes legacy application screens in terms of a terminal application specification and converts the specification into a UML model. (*See* Abstract and Column 1, Lines 57-67) *Stefaniak* repeatedly teaches that the output of the system is a representation or model of the terminal-based application – there is no modification of the terminal-based application itself in *Stefaniak*. (*See* Title; Abstract; Column 1, Lines 15-18 and 28-31) This is further suggested by *Stefaniak*'s continued use of UML, a modeling language, for representing or modeling – as opposed to modifying – the terminal-based application.[1] In other words, even if "the terminal-based application" in *Stefaniak* is comparable to the legacy computer system applications of Claim 1 (which Applicants do not concede), *Stefaniak* fails to disclose, teach, or suggest "automatically

[1] For example, the "Unified Modeling Language (UML) is a language for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modeling and other non-software systems. The UML represents a collection of the best engineering practices that have proven successful in the modeling of large and complex systems." UML specification, available at www.omg.com/uml.

modifying the one or more applications of the legacy computer system that output data" as recited, in part, in Claim 1.

Second, even assuming that "transforming a terminal based screen application into an application specification" could be equated with "automatically modifying the one or more applications of the legacy computer system that output data" (which it cannot, as Applicants discussed above), the "transform[ation of] a terminal based screen application into an application specification," as disclosed in *Stefaniak*, is not "based at least on the mapping of the model of the legacy computer system to the Extensible Markup Language schema," as recited in Claim 1. As discussed above, "transforming a terminal-based screen application" is the first step of the method disclosed in *Stefaniak*. (*See Stefaniak*, Abstract and 1:62-64) It is this first step that the Examiner attempts to equate with "automatically modifying the one or more applications of the legacy computer system that output data," as recited in Claim 1. The Examiner further asserts that the document type definitions of the modeling language-based representation of the application specification, as disclosed in *Stefaniak*, discloses the "mapping of the model of the legacy computer system to the Extensible Markup Language schema," as recited in Claim 1.

Assuming for the sake of argument only that this equation (of the document type definitions with the mapping step recited in Claim 1) is even possible (which Applicants do not concede), how could "transforming a terminal based screen application into an application specification" be equated with "automatically modifying the one or more applications of the legacy computer system that output data [***based on the mapping of the model of the legacy computer system to the Extensible Markup Language schema***]" when the document type definitions of the modeling language-based representation of the application specification in *Stefaniak* (which the Examiner equates with the mapping step recited in Claim 1) do not even exist until the transformation into the application specification, as disclosed in *Stefaniak*, has occurred (in addition to converting the application specification into a UML model of the application specification)? Applicants respectfully submit that it cannot.

Third, *Stefaniak* certainly fails to disclose, teach, or suggest "based at least on the mapping of the model of the legacy computer system to the Extensible Markup Language schema, ***automatically modifying the one or more applications of the legacy computer system***" such that "***the one or more modified applications [are] operable to output data written using a Document Object Model from the legacy computer system in Extensible Markup Language***," as recited in Claim 1. As discussed above, there is no modification of any application in *Stefaniak*; there is merely creation of a model of the legacy applications and then creation of an XML representation ***of that model***. Moreover, there is no modification of any application in *Stefaniak* such that the modified application is operable to output data written using a Document Object Model from the legacy computer system in Extensible Markup Language. Instead, *Stefaniak* discloses "transforming a terminal-based screen application into an application specification," "converting the application specification into a modeling language-based representation" (i.e., the UML model), and "displaying the modeling language based representation [i.e., the UML model] with a graphical user interface." (Abstract) *Stefaniak* merely discloses creating a UML/XML reference model of the UML model (*see Stefaniak*, 6:59-62); the system disclosed in *Stefaniak* does not modify any application such that the modified application is operable to output data written using a Document Object Model from the legacy computer system in Extensible Markup Language, as recited in Claim 1. Even the Title, Abstract, Field of the Invention, and Summary of the Invention sections of *Stefaniak* clearly confirm that *Stefaniak* merely discloses a method for representing terminal-based applications in the unified modeling language, not "automatically modifying the one or more applications of the legacy computer system" such that "the one or more modified applications [are] operable to output data written using a Document Object Model from the legacy computer system in Extensible Markup Language," as recited in Claim 1.

The Examiner acknowledges that *Stefaniak* fails to disclose, in the Examiner's words, "using a Document Object Model." (Office Action, Page 4) However, the Examiner argues that *van Eikeren* teaches "automatically modifying one or more applications of the legacy computer system, the modified application operable to output data written using a Document Object Model from the legacy computer system in Extensible Markup Language." (Office Action, Page 4) Applicants respectfully disagree.

> The cited portion of *van Eikeren* states the following:
>
> Preferably, a standard API, the XML DOM (Document Object Model) is used to access and manipulate XML data. The DOM provides a standard set of properties, methods, and events for program developers to use. This set of standards, the object model, provides a simple means of reading and writing data to and from an XML tree structure.

(*van Eikeren*, 12:5-10; *see* Office Action, Page 4)

This cited portion of *van Eikeren* merely provides its view of what the DOM is and that the DOM may be used (as an example API) to access and manipulate XML data. *Van Eikeren*, however, fails to disclose, teach, or suggest "automatically modifying [based at least on the mapping of the model of the legacy computer system to the Extensible Markup Language schema] the one or more applications of the legacy computer system that output data, the one or more modified applications operable to output data written using a Document Object Model from the legacy computer system in Extensible Markup Language," as recited in Claim 1. Thus, *van Eikeren* clearly fails to make up for at least this deficiency of *Stefaniak*.

As Applicants demonstrated above, even the combination of *Stefaniak* with *van Eikeren* fails to disclose, teach, or suggest each and every limitation recited in Claim 1. Moreover, as discussed below, Applicants respectfully submit that the Examiner has not shown the requisite teaching, suggestion, or motivation in the either *Stefaniak* or *van Eikeren*, or in the knowledge generally available to one of ordinary skill in the art at the time of Applicants' invention, to combine or modify *Stefaniak* and *van Eikeren* in the manner proposed by the Examiner. Claim 1 is allowable for at least this additional reason.

With respect to the proposed *Stefaniak-van Eikeren* combination, the Examiner states, "Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to use DOM, as suggested by van Eikeren, to apply to output XML data. The modification would have been obvious because one of ordinary skill in the art would have been motivated to provide a simple means of reading and writing data to and from an XML tree structure." (Office Action, Page 4) The Examiner's purported motivation is a

statement from *van Eikeren*. (*see van Eikeren*, 12:8-10) However, this statement would not have in any way motivated one of ordinary skill in the art at the time of invention to combine or modify the system disclosed in *Stefaniak* with the system disclosed in *van Eikeren*.[2] In other words, the Examiner's statement does not provide an explanation as to why it would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the particular method for representing terminal-based applications in the UML disclosed in *Stefaniak* with the particular system and techniques disclosed in *van Eikeren*, or how doing so would purportedly meet the limitations of Claim 1. It certainly would not have been obvious to one of ordinary skill in the art at the time of invention to even attempt to, let alone to actually, modify the particular method for representing terminal-based applications in the unified modeling language disclosed in *Stefaniak* in the manner proposed by the Examiner. Applicants respectfully submit that the Examiner's attempt to modify *Stefaniak* appears to constitute the type of impermissible hindsight reconstruction of Applicants' claims, using Applicants' claims as a blueprint, that is specifically prohibited by the M.P.E.P. and governing Federal Circuit cases.

Applicants reiterate the heavy burden incumbent on the Examiner for demonstrating a *prima facie* case of obviousness. As illustrated above, the Examiner's rejection based on the proposed *Stefaniak-van Eikeren* combination does not support a *prima facie* case of obviousness, as is required under the M.P.E.P. and governing Federal Circuit cases.

For at least these reasons, Applicants respectfully request reconsideration and allowance of independent Claim 1 and its dependent claims. For at least certain reasons analogous to those discussed above with reference to independent Claim 1, Applicants respectfully request reconsideration and allowance of independent Claim 4 and its dependent claims.

[2] If "common knowledge" or "well known" art is relied upon by the Examiner to combine or modify the references, Applicants respectfully request that the Examiner provide a reference pursuant to M.P.E.P. § 2144.03 to support such an argument. If the Examiner relies on personal knowledge to supply the required motivation or suggestion to combine or modify the references, Applicants respectfully request that the Examiner provide an affidavit supporting such facts pursuant to M.P.E.P. § 2144.03.

**2. Independent Claim 20 is Allowable**

At a minimum, *Stefaniak*, whether considered alone or in combination with *van Eikeren*, fails to disclose, teach, or suggest the following limitations recited in Claim 20:

- modifying an application of the legacy computer system to output data having a schema element of a target Extensible Markup Language schema, the output data corresponding to a write operation of the application;
- outputting data from the modified application, the output data having the schema element of the target Extensible Markup Language schema;
- aligning the schema element of the output data and a current context;
- writing the schema element of the output data to a current one of plural contexts of the target Extensible Markup Language schema; and
- populating a Document Object Model with the output data to output an Extensible Markup Language instance.

For example, *Stefaniak* fails to disclose, teach, or suggest "modifying an application of the legacy computer system to output data having a schema element of a target Extensible Markup Language schema, the output data corresponding to a write operation of the application," as recited in Claim 20. *Stefaniak* is directed to a method for representing terminal-based applications in the UML. (*Stefaniak*, 1:15-17) In particular, *Stefaniak* merely discloses generating models of legacy applications - it is not modifying legacy computer applications, let alone "modifying an application of the legacy computer system to output data having a schema element of a target Extensible Markup Language schema, the output data corresponding to a write operation of the application," as recited in Claim 20.

As allegedly disclosing this element of Claim 20, the Examiner references Figure 6 of *Stefaniak*. (Office Action, Page 8) Figure 6 of *Stefaniak* merely discloses "a flowchart depicting a process of generating an XML file representation of the UML model created by the process described in FIG. 5A through 5C [of *Stefaniak*]." (*Stefaniak*, 6:59-62) The UML model is a model of an application specification that was created from a terminal-based application. Thus, Figure 6 of *Stefaniak* discloses generating an XML file representation of a UML model of an application specification, the application specification having been created from a terminal-based application (which the Examiner equates with the legacy computer system application). Figure 6 of *Stefaniak* clearly fails to disclose, teach, or suggest "modifying an application of the legacy computer system to output data having a schema element of a target Extensible Markup Language schema, the output data corresponding to a

write operation of the application," as recited in Claim 20. In other words, the terminal-based application disclosed in *Stefaniak* is not modified such that it outputs data having a schema element of a target Extensible Markup Language schema, the output data corresponding to a write operation of the application, as recited in Claim 20.

Additionally, *Stefaniak* discloses a system that describes legacy application screens in terms of a terminal application specification and converts the specification into a UML model. (*See* Abstract and Column 1, Lines 57-67) *Stefaniak* repeatedly teaches that the output of the system is a representation or model of the terminal-based application – there is no modification of the terminal-based application itself in *Stefaniak*. (*See* Title; Abstract; Column 1, Lines 15-18 and 28-31) This is further suggested by *Stefaniak*'s continued use of UML, a modeling language, for representing or modeling – as opposed to modifying – the terminal-based application. In other words, even if "the terminal-based application" in *Stefaniak* is comparable to the legacy computer system applications of Claim 20 (which Applicants do not concede), *Stefaniak* fails to disclose, teach, or suggest "modifying an application of the legacy computer system to output data having a schema element of a target Extensible Markup Language schema, the output data corresponding to a write operation of the application" as recited, in part, in Claim 20.

*Stefaniak* merely discloses creation of a model of the legacy applications and then creation of an XML representation of that model. *Stefaniak* discloses "transforming a terminal-based screen application into an application specification," "converting the application specification into a modeling language-based representation" (i.e., the UML model), and "displaying the modeling language based representation [i.e., the UML model] with a graphical user interface." (Abstract) *Stefaniak* merely discloses creating a UML/XML reference model of the UML model (*see Stefaniak*, 6:59-62); the system disclosed in *Stefaniak* does not modify any application such that the modified application outputs data having a schema element of a target Extensible Markup Language schema, the output data corresponding to a write operation of the application, as recited in Claim 20. Even the Title, Abstract, Field of the Invention, and Summary of the Invention sections of *Stefaniak* clearly confirm that *Stefaniak* merely discloses a method for representing terminal-based applications in the unified modeling language, not "modifying an application of the legacy computer

system" such that the modified application outputs "data having a schema element of a target Extensible Markup Language schema, the output data corresponding to a write operation of the application," as recited in Claim 20.

As another example, *Stefaniak* fails to disclose, teach, or suggest "outputting data from the modified application, the output data having the schema element of the target Extensible Markup Language schema," as recited in Claim 20. First, at least because *Stefaniak* fails to disclose, teach, or suggest "modifying an application of the legacy computer system to output data having a schema element of a target Extensible Markup Language schema, the output data corresponding to a write operation of the application," as recited in Claim 20, *Stefaniak* necessarily fails to disclose, teach, or suggest "outputting data from the modified application, the output data having the schema element of the target Extensible Markup Language schema," as recited in Claim 20. Second, *Stefaniak* merely discloses generating an XML file representation of a UML model of an application specification created from a terminal-based application. The terminal-based application (which the Examiner equates with the legacy computer system application recited in Claim 20) is not modified to output data having a schema element of an XML schema and thus, does not disclose, teach, or suggest "outputting data from the modified application, the output data having the schema element of the target Extensible Markup Language schema," as recited in Claim 20.

Moreover, even assuming for the sake of argument only that the application specification disclosed in *Stefaniak* could be equated with the modified legacy computer system application recited in Claim 20 (with which Applicants disagree), the application specification does not "output data having a schema element of a target Extensible Markup Language schema, the output data corresponding to a write operation of the application," as recited in Claim 20. Instead, another component of the system disclosed in *Stefaniak* creates a UML model of the application specification and then generates an XML file representation of the UML model of the application specification.

Applicants respectfully submit that *Stefaniak* fails to disclose, teach, or suggest at least certain the remaining limitations of Claim 20; however, to avoid burdening the record

and in view of the clear distinctions discussed above, Applicants do not specifically discuss each of these distinctions in this Response.

The Examiner acknowledges that *Stefaniak* fails to disclose, in the Examiner's words, "using a Document Object Model." (Office Action, Page 9) However, the Examiner argues that *van Eikeren* teaches "populating a Document Object Model with the output data to output an Extensible Markup Language instance." (Office Action, Page 9) Applicants respectfully disagree.

The cited portion of *van Eikeren* states the following:

> Preferably, a standard API, the XML DOM (Document Object Model) is used to access and manipulate XML data. The DOM provides a standard set of properties, methods, and events for program developers to use. This set of standards, the object model, provides a simple means of reading and writing data to and from an XML tree structure.

(Column 12, Lines 5-10; *see* Office Action, Page 4)

This cited portion of *van Eikeren* merely provides its view of what the DOM is and that the DOM may be used (as an example API) to access and manipulate XML data. *Van Eikeren*, however, fails to disclose, teach, or suggest "populating a Document Object Model with the output data to output an Extensible Markup Language instance," as recited in Claim 20. Thus, *van Eikeren* clearly fails to make up for at least this deficiency of *Stefaniak.*

As Applicants demonstrated above, even the combination of *Stefaniak* with *van Eikeren* fails to disclose, teach, or suggest each and every limitation recited in Claim 20. Moreover, Applicants respectfully submit that the Examiner has not shown the requisite teaching, suggestion, or motivation in the either *Stefaniak* or *van Eikeren*, or in the knowledge generally available to one of ordinary skill in the art to combine or modify *Stefaniak* and *van Eikeren* in the manner proposed by the Examiner. Claim 20 is allowable for at least this additional reason.

With respect to the proposed *Stefaniak-van Eikeren* combination, the Examiner states, "Therefore, it would have been obvious to one of ordinary skill in the art at the time the

invention was made to use DOM, as suggested by van Eikeren, to apply to output XML data. The modification would have been obvious because one of ordinary skill in the art would have been motivated to provide a simple means of reading and writing data to and from an XML tree structure." (Office Action, Page 4) For at least those reasons discussed above with reference to Claim 1, Applicants respectfully submit that the Examiner has not demonstrated the requisite teaching, suggestion, or motivation in the cited references or in the knowledge generally available to one of ordinary skill in the art at the time of Applicants' invention to combine or modify the references in the manner the Examiner proposes. Thus, Applicants respectfully submit that the Examiner's rejection based on the proposed *Stefaniak-van Eikeren* combination does not support a *prima facie* case of obviousness, as is required under the M.P.E.P. and governing Federal Circuit cases.

For at least these reasons, Applicants respectfully request reconsideration and allowance of independent Claim 20 and its dependent claims.

### B. Independent Claim 13 is Allowable over the Proposed *Lection-Stefaniak* Combination

The Examiner rejects Claims 13-14 under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent 6,418,446 to Lection, et al. ("*Lection*") in view of *Stefaniak*. Applicants respectfully disagree and discuss independent Claim 13 as an example.

*Lection*, whether considered alone or in combination with *Stefaniak*, fails to disclose, teach, or suggest at least the following limitations as recited in Claim 13:

- a computer system having an application that outputs data, each data output instance corresponding to a write operation of the application;
- a writer engine loaded on the computer system and interfaced with the application, the writer engine having an Extensible Markup Language schema as a data file and the writer engine operable to write the data output by the application in plural active contexts;
- wherein the application calls the writer engine when the application outputs data, the writer engine operable to build a Document Object Model instance for output of the data in accordance with the Extensible Markup Language schema.

*Lection* does not even relate to "outputting data from a Document Object model as Extensible Markup Language," as recited in Claim 13. Instead, *Lection* is directed to a process, system, and method for gathering data having dynamically variable record formats such as those created when a dynamic schema is used with a data repository. (Column 3, Lines 30-34) In other words, *Lection* is directed to mapping ***stored data records, which are already structured in XML format***, to variable record formats. The Examiner appears to equate "the source data" of *Lection*, which is stored as a structure data record, with "the output data" in Claim 13. (*See* Office Action, Page 10) Applicants respectfully submit that *Lection* does not support this interpretation. The source data disclosed in *Lection* is merely ***a stored structured data record, it is not data output from an application running on a computer system, and it does not correspond to a write operation of the application*** as recited in Claim 13. Thus, *Lection* fails to disclose, teach, or suggest "a computer system having an application that outputs data, each data output instance corresponding to a write operation of the application," as recited in Claim 13.

The Examiner correctly acknowledges that *Lection* fails to disclose a writer engine, as recited in Claim 13. However, the Examiner argues that *Stefaniak* teaches, as stated by the Examiner, "an engine operable to write that data output by the application in plural active contexts; wherein the application calls the writer engine when the application outputs data, the writer engine operable to build a Document Object Model instance for output of the data in accordance with the Extensible Markup Language schema." (Office Action, Page 10) Applicants respectfully disagree.

As purportedly disclosing these limitations, the Examiner references the following portion of *Stefaniak*:

> The terminal screens are discovered using the transform navigator 19, which produces application and screen specifications 32. The application and screen specifications 32 are then applied to the file warehouse 21, which produces the project file reference model 27. The model 27 is applied to the terminal-to-XML 20, which produces a UML model 34 in a MOF compliant repository. The UML model is then applied to the UML model to XMI/UML DTD generator 22, which produces XMI/UML DTD streams 35. The streams 35 may be used for several purposes, including transmitting legacy screen based application specifications over a network to modeling tools. From the

> modeling tools the application specifications could be viewed in an object oriented way compliant with the UML standard.

(*Stefaniak*, 5:43-57; Office Action, Page 10)

However, nowhere does this cited portion disclose, teach, or suggest "a writer engine loaded on the computer system and interfaced with the application, the writer engine having an Extensible Markup Language schema as a data file and the writer engine operable to write the data output by the application in plural active contexts" or "wherein the application calls the writer engine when the application outputs data, the writer engine operable to build a Document Object Model instance for output of the data in accordance with the Extensible Markup Language schema," as recited in Claim 13. Instead, the cited portion discloses that a transform navigator creates application and screen specifications of a terminal-based application. Then, a project file reference model is created from the application and screen specifications. Next, a terminal-to-XML component creates a UML model of the project file reference model (created from the application and screen specifications). Next, an XMI/UML DTD generator produces *from the UML model* XMI/UML DTD streams.

Apparently, the Examiner equates the XMI/UML DTD generator with the writer engine recited in Claim 13. (Office Action, Page 10) However, the XMI/UML DTD generator merely generates UML/XML representations of the UML model. The cited portion of *Stefaniak* does not disclose, teach, or suggest that the XMI/UML DTD generator "ha[s] an Extensible Markup Language schema as a data file" or is operable to "write the data output by the application in plural active contexts," as recited in Claim 13.

Additionally, neither *Stefaniak* nor *Lection* discloses, teaches, or suggests "wherein the application calls the writer engine when the application outputs data, the writer engine operable to build a Document Object Model instance for output of the data in accordance with the Extensible Markup Language schema," as recited in Claim 13. First, *Stefaniak* fails to even mention the Document Object Model. Thus, *Stefaniak* (which is the only reference cited by the Examiner as allegedly disclosing this limitation) necessarily fails to disclose, teach, or suggest "the writer engine operable to build a Document Object Model instance for output of the data in accordance with the Extensible Markup Language schema," as recited in

Claim 13. Furthermore, neither *Stefaniak* nor *Lection* discloses, teaches, or suggests that an application that calls the writer engine when the application outputs data, as recited in Claim 13.

As Applicants demonstrated above, even the combination of *Lection* with *Stefaniak* fails to disclose, teach, or suggest each and every limitation recited in Claim 13. Moreover, Applicants respectfully submit that the Examiner has not shown the requisite teaching, suggestion, or motivation in the either *Stefaniak* or *Lection*, or in the knowledge generally available to one of ordinary skill in the art at the time of Applicants' invention, to combine or modify *Stefaniak* and *Lection* in the manner proposed by the Examiner. Claim 13 is allowable for at least this additional reason.

With respect to the proposed *Lection-Stefaniak* combination, the Examiner states, "Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate XMI/UML DTD generator, as suggested by Stefaniak into the system of Lection, to produce XMI/UML DTD streams. The modification would have been obvious because one of ordinary skill in the art would have been motivated to provide a simple means to generate XML codes." (Office Action, Page 10) This statement would not have in any way motivated one of ordinary skill in the art at the time of invention to combine or modify the system disclosed in *Stefaniak* with the system disclosed in *Lection*.[3] In other words, the Examiner's statement does not provide an explanation as to why it would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the particular method for representing terminal-based applications in the unified modeling language disclosed in *Stefaniak* with the particular system and techniques disclosed in *Lection*, or how doing so would purportedly meet the limitations of Claim 13. It certainly would not have been obvious to one of ordinary skill in the art at the time of invention to even attempt to, let alone to actually, modify the particular method for representing terminal-based applications in the unified modeling language disclosed in *Stefaniak* in the manner proposed by the Examiner. Applicants respectfully submit that the Examiner's attempt to

[3] If "common knowledge" or "well known" art is relied upon by the Examiner to combine or modify the references, Applicants respectfully request that the Examiner provide a reference pursuant to M.P.E.P. § 2144.03 to support such an argument. If the Examiner relies on personal knowledge to supply the required motivation or suggestion to combine or modify the references, Applicants respectfully request that the Examiner provide an affidavit supporting such facts pursuant to M.P.E.P. § 2144.03.

modify *Stefaniak* appears to constitute the type of impermissible hindsight reconstruction of Applicants' claims that is specifically prohibited by the M.P.E.P. and governing Federal Circuit cases.

Accordingly, since the references fail to provide the required teaching, suggestion, or motivation to modify *Stefaniak* in the manner the Examiner proposes, Applicants respectfully submit that the Examiner's conclusions set forth in the Office Action fall short of the requirements set forth in the M.P.E.P. and the governing Federal Circuit case law for demonstrating a *prima facie* case of obviousness. Thus, Applicants respectfully submits that the Examiner's proposed modifications to *Stefaniak* (and the proposed combination of *Lection* with *Stefaniak*) appears to be merely an attempt, with the benefit of hindsight, to reconstruct Applicants' claims and is unsupported by the teachings of *Stefaniak* and *Lection*.

Applicants reiterate the heavy burden incumbent on the Examiner for demonstrating a *prima facie* case of obviousness. As illustrated above, the Examiner's rejection based on the proposed *Lection-Stefaniak* combination does not support a *prima facie* case of obviousness, as is required under the M.P.E.P. and governing Federal Circuit cases.

For at least these reasons, Applicants respectfully request reconsideration and allowance of independent Claim 13 and its dependent claims.

**C. Dependent Claims 3, 15-19, and 22 are Allowable**

The Examiner rejects Claim 3 under 35 U.S.C. § 103(a) as being unpatentable over *Stefaniak* in view of *van Eikeren* and U.S. Patent 6,347,307 to Sandhu ("*Sandhu*"). The Examiner rejects Claims 15-18 under 35 U.S.C. § 103(a) as being unpatentable over *Lection* in view of *Stefaniak* and Shanmugasundaram, et al. *Relational Databases for Querying XML Documents: Limitations and Opportunities* Proceedings of the 25th VLDBConference, Edinburgh, Scotland, 1999 ("*Shanmugasundaram*"). The Examiner rejects Claim 19 under 35 U.S.C. § 103(a) as being unpatentable over *Lection* in view of *Stefaniak*, *Shanmugasundaram*, and U.S. Patent 6,209,124 to Vermeire ("*Vermeire*"). The Examiner rejects Claim 22 under 35 U.S.C. § 103(a) as being unpatentable over *Stefaniak* in view of *van Eikeren* and *Shanmugasundaram*. Applicants respectfully disagree.

Dependent Claims 3, 15-19, and 22 depend from independent Claims 1, 13, and 20, respectively, which Applicants have shown above to be allowable, and are allowable for at least this reason. Additionally, dependent Claims 3, 15-19, and 22 recite further patentable distinctions over the references cited in the Examiner's rejections. To avoid burdening the record and in view of the clear allowability of independent Claims 1, 13, and 20, Applicants do not specifically discuss these distinctions in this Response; however, Applicants reserve the right to discuss these distinctions in a future Response or on Appeal, if appropriate. Furthermore, with respect to all of the proposed combinations of references made by the Examiner, Applicants do not admit that the proposed combinations of references are possible or that the Examiner has demonstrated the required teaching, suggestion, or motivation in the cited references or in the knowledge generally available to one of ordinary skill in the art the time of Applicants' invention to combine or modify these references in the manner proposed.

For at least these reasons, Applicants respectfully request reconsideration and allowance of Claims 3, 15-19, and 22.

## III. No Waiver

All of Applicants' arguments and amendments are without prejudice or disclaimer. Additionally, Applicants have merely discussed example distinctions from the various references cited by the Examiner. Other distinctions may exist, and Applicants reserve the right to discuss these additional distinctions in a later Response or on Appeal, if appropriate. By not responding to additional statements made by the Examiner, Applicants do not acquiesce to the Examiner's additional statements. The example distinctions discussed by Applicants are sufficient to overcome the Examiner's rejections.

**Conclusion**

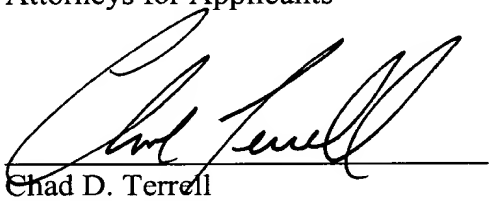
Applicants have made an earnest attempt to place this case in condition for immediate allowance. For at least the foregoing reasons, Applicants respectfully request allowance of all pending claims.

If the Examiner feels that prosecution of the present Application may be advanced in any way by a telephone conference, the Examiner is invited to contact the undersigned attorney at 214.953.6813.

The Commissioner is authorized to charge the amount of $120.00 to cover the one-month extension-of-time fee to Deposit Account No. 05-0765 of Electronic Data Systems Corporation. Although no other fees are believed to be due, the Commissioner is hereby authorized to charge any additional fees or to credit any overpayment to Deposit Account No. 05-0765 of Electronic Data Systems Corporation.

Respectfully submitted,

BAKER BOTTS L.L.P.
Attorneys for Applicants



Chad D. Terrell
Reg. No. 52,279

Date: April 22, 2005

Customer Number: **35005**